An Optimization Algorithm Based On Stochastic Sensitivity Analysis For Noisy Objective Landscapes (5th Workshop on Stochastic Numerics)

AUTHOR(S):

Okano, Hiroyuki; Koda, Masato

数理解析研究所講究録 1240 巻 2001 年 47-57

An Optimization Algorithm Based On Stochastic Sensitivity Analysis For Noisy Objective Landscapes

日本 IBM・東京基礎研究所　岡野 裕之　(Hiroyuki Okano)
IBM Research, Tokyo Research Laboratory
E-mail: okanoh@jp.ibm.com

筑波大・社会工学系　香田 正人　(Masato Koda)
Institute of Policy and Planning Sciences, Univ. of Tsukuba
E-mail: koda@shako.sk.tsukuba.ac.jp

Abstract

A function minimization algorithm such that a solution is updated based on derivative information approximated with sample points is proposed. The algorithm generates sample points with Gaussian white noise, and approximates derivatives based on stochastic sensitivity analysis. Unlike standard trust region methods which calculate gradients with n or more sample points, where n is the number of variables, the proposed algorithm allows the number of sample points M to be less than n. Furthermore, it ignores small amounts of noise within a trust region. This paper addresses the following two questions: To what extent does the derivative approximation become worse when the number of sample points is small? Does the algorithm converge to a good solution with inexact derivative information when the objective landscape is noisy? Through intensive numerical experiments using quadratic functions, the algorithm is shown to be able to approximate derivatives when M is about $n/10$ or more. The experiments using a formulation of the traveling salesman problem (TSP) shows that the algorithm can find reasonably good solutions for noisy objective landscapes with inexact derivatives.

1. Introduction

Optimization problems that seek for minimization (or equivalently maximization) of an objective function have practical importance in various areas. Once a task is modeled as an optimization problem, general optimization techniques become applicable; e.g., linear programming, gradient methods, etc. One may encounter difficulties, however, in applying these techniques when the objective function is non-differentiable or it is defined as a procedure. The example problem considered in this paper involving such a function is a parametric local search for the *t*raveling salesman problem (TSP), a representative combinatorial optimization problem, in which an objective function of a

parameter vector is defined as a heuristic procedure. Another example, which has been studied by the authors [7] but which is not covered in this paper, is a model involving a step function. For both cases, the associated objective landscapes are noisy in the sense that they include many non-differentiable points and many local minima.

In this paper, an unconstrained function minimization algorithm such that a solution is updated based on derivative approximated by random sampling is proposed for such noisy landscapes. The assumption of this study is that the number of sample points may be less than the dimension n of the objective function. Note that standard trust region methods require n or more sample points (see [1,2,3,4,5] and references therein). For example, direct search methods [2,3] maintain $n+1$ non-degenerate points within a trust region, and search methods that use quadratic polynomial interpolation require $(n+1)(n+2)/2$ non-degenerate points. The questions arising here are:

- To what extent does the derivative approximation become worse when the number of sample points is small?
- Does the algorithm converge to a good solution with inexact derivative information when the objective landscape is noisy?

This paper aims to answer these questions through numerical experiments.

The optimization algorithm proposed in this paper assumes that the objective function is scaled such that an area formed by the Gaussian distribution of unit variance can be used as a trust region, and uses stochastic sensitivity analysis to approximate derivatives. The method is to inject Gaussian white noise into each of the variables in the current solution, and apply Novikov's theorem [6] to obtain sensitivities (i.e., gradients) of the variables.

The paper is organized as follows: In Section 2, the optimization algorithm is described, and a new simple derivation of Novikov's theorem is presented. Section 3 shows numerical experiments to answer the first question. In Section 4, the algorithm is applied to the TSP which has a noisy objective landscape to answer the second question. Finally, Section 5 summarizes the paper.

2. Stochastic gradient method

Consider an unconstrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \in \mathbb{R} \\ \text{subject to} & x \in \mathbb{R}^n. \end{array} \tag{1}$$

Gradient methods iteratively update the current solution x as

$$x \leftarrow x + \mu \frac{\delta x}{|\delta x|}, \quad \delta x = -\nabla f(x), \tag{2}$$

Algorithm framework of SNR

1. Initialize the current solution $x := x^0$.
2. Initialize the best solution $x^{best} := x$.
3. **For** $k := 1, 2, \ldots, N$ **do begin** // N iterations.
4. Initialize decent direction $\delta x := 0$.
5. **For** $j := 1, 2, \ldots, M(= 100)$ **do begin** // Derivative approximation.
6. Generate a noise vector ξ^j. // $\xi_i^j \in N(0,1)$, $i = 1, 2, \ldots, n$.
7. $\delta x_i := \delta x_i - f(x^j)\xi_i^j$. // $\langle f(x(\xi))\xi_i \rangle = \langle \frac{\partial f(x)}{\partial x_i} \rangle$.
8. **If** $f(x^{best}) > f(x^j)$ **then** $x^{best} := x^j$. // Save the best solution.
9. **end;**
10. $w := \max_i |\delta x_i|$. // Find max component in δx.
11. $s := \arg\min_{s=1,2,\ldots,100} f(x + 0.01s\frac{\delta x}{w})$. // Line search by sampling.
12. $x := x + \mu \frac{\delta x}{|\delta x|}$ where $\mu = 0.01s\frac{|\delta x|}{w}$. // Update the current solution.
13. **If** $f(x^{best}) > f(x)$ **then** $x^{best} := x$. // Save the best solution.
14. **If** terminal condition is met **then goto** 16.
15. **end;**
16. Output x^{best}.

Figure 1. Algorithm framework of SNR.

where μ is a step width determined by line search, δx is a descent direction, and $\nabla f(x)$ is a gradient vector defined by

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)^T. \tag{3}$$

To avoid the exact gradient calculation of (3), and to cope with noisy objective landscapes, Koda and Okano proposed a noise-based gradient method for artificial neural network learning [7], and further modified the method for function minimization [8]. The algorithm, called *s*tochastic noise reaction (SNR), injects a Gaussian white noise sequence with zero mean and unit variance, $\xi_i \in N(0,1)$, into a variable x_i as

$$x_i^j = x_i + \xi_i^j, \tag{4}$$

where ξ_i^j denotes the j-th noise in the noise sequence injected into the i-th variable. Each component of a derivative is approximated without explicitly differentiating the objective function by using

$$\langle \frac{\partial f(x)}{\partial x_i} \rangle = \frac{1}{M} \sum_{j=1}^{M} f(x^j)\xi_i^j, \tag{5}$$

where $\langle \cdot \rangle$ denotes the expectation operator, and M is a loop count for taking the average. Note that, in Eq. (5), all the components in the gradient $\nabla f(x)$, i.e., $\frac{\partial f(x)}{\partial x_i}$, $i = 1, 2, \ldots, n$,

are computed at the same time, which means the gradient approximation requires the objective function to be evaluated M times, so that the dimension n does not explicitly dominate the computational overhead. The value of M was set to 100 in all of the numerical experiments here.

In this paper, SNR is used within the algorithm framework described in Fig. 1. A noise sequence ξ_i for each variable x_i is formally generated in Step 6, while, in actual implementations, all the noise sequences should be generated and normalized in advance to ensure $\langle \xi_i \rangle = 0$. In Step 11, the next solution is searched for using line search by sampling, in which the maximum displacement of the sampling point farthest from the current solution is 1. When more than two solutions on the line $x + 0.01s\frac{\delta x}{w}$, $s = 1, 2, \ldots, 100$, share the same minimum value, the one with larger value of s is selected so that the search does not stay within a small region.

2.1. Novikov's theorem

Equation (5) relies on the following identity (Novikov's theorem [6]):

$$\langle \frac{\delta H(\xi)}{\delta \xi_i} \rangle = \langle H(\xi)\xi_i \rangle, \tag{6}$$

where $H(\xi)$ is an arbitrary function of Gaussian stochastic sequences ξ_i, $i = 1, 2, \ldots, n$, and $\frac{\delta H(\xi)}{\delta \xi_i}$ denotes the functional derivative [9]. ξ_i is a Gaussian white noise with zero mean and unit variance; i.e.,

$$\langle \xi_i \rangle = 0, \quad \langle \xi_i^t \xi_j^s \rangle = \delta_{ij}\delta_{ts}, \tag{7}$$

where δ_{ij} and δ_{ts} denote the Kronecker delta, and ξ_i^t denotes the t-th noise in the noise sequence injected into the i-th variable.

The derivation of the theorem given in [6] is lengthly, but the same result is obtained using integration by parts as follows:

$$\begin{aligned} \langle \tfrac{\delta H(\xi)}{\delta \xi_i} \rangle &= \int \tfrac{\delta H(\xi)}{\delta \xi_i} G(\xi) d\xi \\ &= -\int H(\xi) \tfrac{\delta G(\xi)}{\delta \xi_i} d\xi + \int \tfrac{\delta}{\delta \xi_i} \{ H(\xi)G(\xi) \} d\xi \\ &= \tfrac{1}{\sigma_i^2} \int H(\xi)\xi_i G(\xi) d\xi \\ &= \tfrac{1}{\sigma_i^2} \langle H(\xi)\xi_i \rangle, \end{aligned} \tag{8}$$

where $H(\xi)$ is a real valued smooth functional with polynomial growth at infinity, and the Gaussian kernel $G(\xi)$ is defined as

$$G(\xi) = \frac{\exp\left(-\int_{-\infty}^{+\infty} \sum_i \frac{\xi_i^2(t)}{2\sigma_i^2} dt\right)}{\left\langle \exp\left(-\int_{-\infty}^{+\infty} \sum_i \frac{\xi_i^2(t)}{2\sigma_i^2} dt\right) \right\rangle}. \tag{9}$$

Note that $\sigma_i = 1$ is assumed in Eqns. (5) and (6). Instead of using plain integration by parts formula (8), we may note that the analogous result can be derived using the integration by parts of the Malliavin calculus (e.g., see the article by Kohatsu-Higa in this report).

3. Performance of derivative approximation

Figure 2 shows the curve of $f(x) = x^2 + 10\cos(10x)$ and its derivatives approximated using Novikov's theorem. The second term of the function, $10\cos(10x)$, is meant to be noise. The figure shows that the derivative $2x$ is approximated while the noise is ignored. Note that the finite difference method is not usable when noise exists. Figure 3 shows derivatives of the same function approximated by a method based on Simulated Annealing (SA) [10]. Simulated annealing, as well as SNR, is a stochastic method, and is able to escape from local minima; i.e., it can cope with noise. The method used in Fig. 3 approximates derivatives such that it performs a 10-step random walk from a current solution x to obtain $\tilde{x}$, and calculates $(f(\tilde{x}) - f(x))/(\tilde{x} - x)$. In the random walk, uphill moves from x to x', $d = f(x') - f(x) > 0$, are accepted with probability $\exp(-d/T)$, where a pseudo temperature parameter T is set to $-1.0/\log(0.5)$. The figure shows the SA-based method can ignore the noise and approximate derivatives as well as the method using Novikov's theorem. (This is only true for objective functions having few dimensions.)

Figure 4 shows the maximum, the average, and the minimum values of inner angles between approximated derivatives and exact values for the n dimensional quadratic function $f(x) = (1/n)\sum_{i=1}^{n} x_i^2$ at $x_i = 10.0$ over 100 trials. The figure shows the angles are less than 45° on the average when $n < M$, i.e., the number of sample points (length of the noise sequence) $M = 100$ is greater than the dimension n, and they approach 90° as n increases. Note that random vectors may have values greater than 90° (Fig. 7), and the result in Fig. 4 is significantly better than random vectors.

Figure 5 shows the same plot as in Fig. 4 but using the method based on SA with a 100-step random walk. The figure shows the method fails to approximate the derivatives when $n > 10$, which is less than the number of random walk steps. This implies that SA's performance is not acceptable when the dimension is high, and, given the same number of sample points and random walk steps, the derivative approximation using Novikov's theorem performs better than SA. Figure 6 shows the same plot as in Fig. 4 using a method based on the direct search method [2,3], in which $M = 100$ points are sampled using Gaussian white noise, and the gradient direction is approximated by $x_{max} - x_{ave}$ with the maximum point x_{max} and their average x_{ave}. The performance of

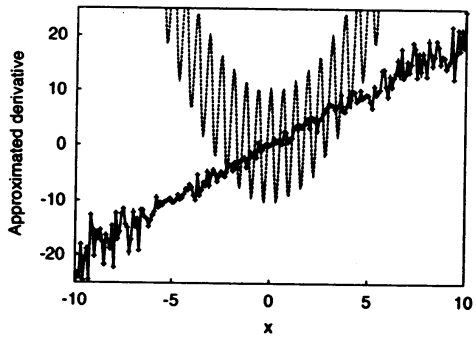

Figure 2. Derivative approximation for a one-dimensional quadratic function with noise using Novikov's theorem.

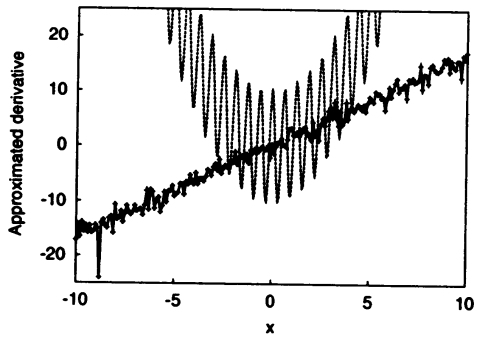

Figure 3. Derivative approximation for a one-dimensional quadratic function with noise using simulated annealing.

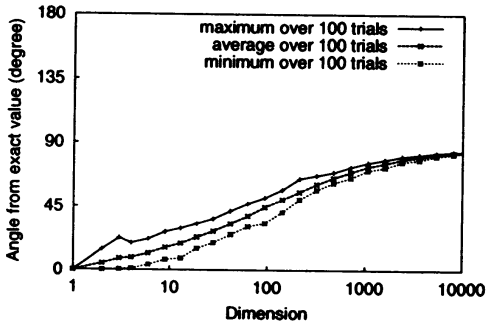

Figure 4. Derivative approximation for a multi-dimensional quadratic function using Novikov's theorem.

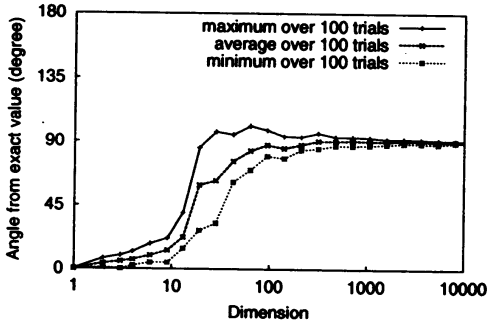

Figure 5. Derivative approximation for a multi-dimensional quadratic function using simulated annealing.

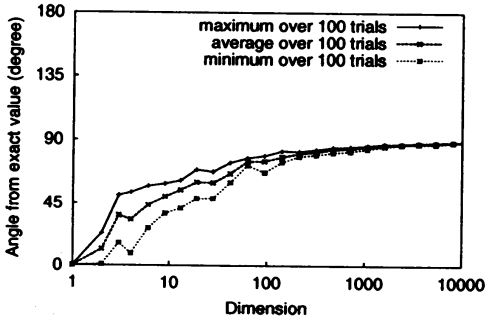

Figure 6. Derivative approximation for a multi-dimensional quadratic function using a direct search method.

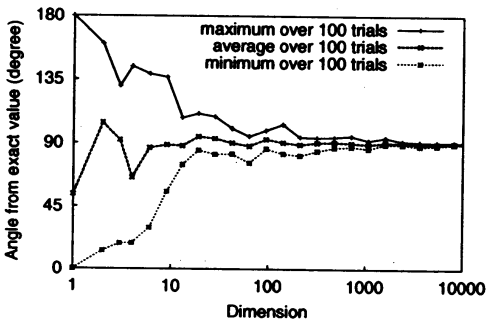

Figure 7. Difference between derivatives of a multi-dimensional quadratic function and random vectors.

this method is similar to but not better than one using Novikov's theorem (Fig. 4).

The numerical experiments in this section show that when the number of sample points generated at each iteration is less than n, or when the objective landscape is noisy, the derivative approximation using Novikov's theorem is a reasonable choice. A drawback of this algorithm, however, is that it requires generating all the sample points at each iteration, while other trust region methods allow retaining the sample points within a trust region and reusing them. To overcome this drawback, new stochastic sensitivity analysis techniques should be developed.

4. Application to combinatorial optimization

In this section, a TSP formulation based on an addition heuristic is proposed, and SNR is applied to it. The formulation involves a procedure, so that the derivative is not available. Moreover, no subgradient for this formulation has yet been identified.

4.1. The traveling salesman problem

The traveling salesman problem (TSP) is a representative NP-hard combinatorial optimization problem, and is known to have a wide range of practical applications. In the TSP, a set of vertices $V = \{1, 2, \ldots, |V|\}$ and a distance between each pair of vertices i and j, d_{ij}, are given, and the problem is to find an ordering π of vertices that minimizes a tour length defined by

$$h(\pi) = \sum_{i=1}^{|V|} d_{\pi(i),\pi(i+1)}, \tag{10}$$

where the index of π is defined modulo $|V|$ so that vertex $\pi(|V|)$ is adjacent in the tour to both $\pi(|V|-1)$ and $\pi(1)$. Note that here the geometric TSP is assumed, which means the vertices are mapped on a plane, and the distance is Euclidean. The objective function of the TSP, i.e., (10), takes a discrete vector π, and thus gradient methods cannot be applied directly. One of the best known heuristics for the TSP is the k-opt heuristic by Lin and Kernighan (LK) [11] that produces tours whose lengths are 1 to 2 percent in excess of optimal. LK is used for comparison in the numerical experiments in Subsection 4.4.

4.2. The addition heuristic

The addition heuristic, starting from a subtour consisting of a single vertex, inserts vertices one by one into the place in the subtour that least increases the tour length. An ordering of vertices, $a(i)$, $i = 1, 2, \ldots, |V|$, to insert into the subtour is called an *addition*

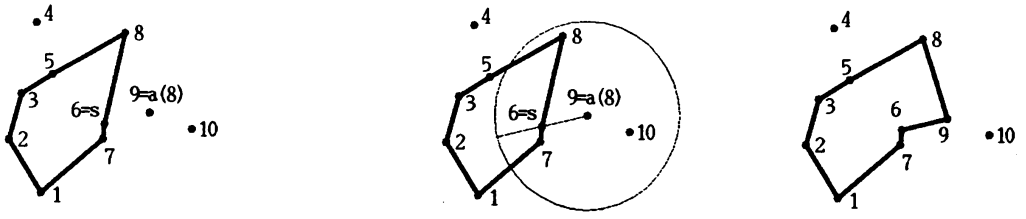

Figure 8. An example of the insertion procedure in the addition heuristic. (a) A subtour. The next vertex to be inserted into the subtour is 9. (b) Candidates for insertion positions – $(1,7)$, $(7,6)$, $(6,8)$, and $(8,5)$ – are identified. (c) Vertex 9 is inserted between vertices 6 and 8.

sequence. The offline addition heuristic, to which a is given in advance, is defined as follows:

*O*ffline addition heuristic $AH(a)$

1. Let $a(1)$ be a subtour consisting of a single vertex.
2. **For** $i = 2, 3, \ldots, |V|$ **do begin**
3. Let s be the nearest vertex in the subtour from $a(i)$ (Fig. 8 (a)).
4. Let $W(s, a(i))$ be a set of vertices in the subtour inside the circle of radius $2d_{s,a(i)}$ centered at $a(i)$ (Fig. 8 (b)).
5. Find the place in the subtour, either before or after $t \in W(s, a(i))$, that least increases the tour length when $a(i)$ is inserted into the place (Fig. 8 (c)).
6. Cut the edge found in Step 5, and insert $a(i)$ at that location.
7. **end;**
8. Output $\pi(i)$, $i = 1, 2, \ldots, |V|$, the ordering of vertices in the resulting tour.

It is not known that for every TSP instance there always exists at least one addition sequence with which AH finds an optimal solution, however, we conjecture it is true for the Euclidean TSP. (When the search radius used in Step 4 is $d_{s,a(i)}$, one can find a counter example.) In AH, local changes in the input addition sequence do not greatly change the resulting tour. For example, when a subtour consists of $|V| - 2$ vertices, insertions of $a(|V| - 1)$ and $a(|V|)$ into the subtour can be performed independently in most cases. Overall positions (or priorities) of the vertices in a, on the other hand, do affect the quality of the resulting tour. Based on this observation, we conjecture that a good tour can be obtained if key vertices which characterize the optimal tour have higher priorities in the addition sequence.

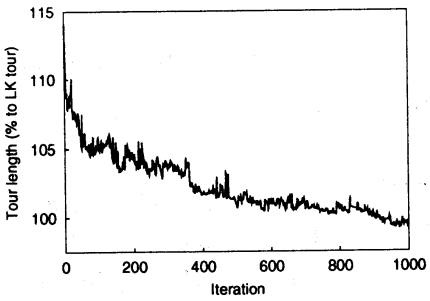

Figure 9. Convergence behavior of SNR for the TSP.

4.3. Addition heuristic-based objective function

The addition heuristic $AH(a)$ described in the last subsection cannot be used directly in gradient methods because it requires a discrete vector a. To relax the discrete property, an n-dimensional real vector x is introduced where $n = |V|$. Each component x_i specifies the priority of a corresponding vertex i in an addition sequence; i.e., an addition sequence is generated by sorting x_i in decreasing order so that $x_{a(i)} \geq x_{a(i+1)}$. Then, the addition heuristic-based formulation is defined as follows:

$$f(x) = h(AH(decreasingorder(x))), \tag{11}$$

where $decreasingorder(x)$ maps the priority vector x to an addition sequence a, $AH(a)$ generates an ordering π by using the addition heuristic, and $h(\pi)$ computes the corresponding tour length. Note that the objective function of this formulation is normally defined as a subroutine in a computer program.

4.4. Numerical experiments

SNR was applied to a random instance with $|V| = 1000$ on a unit square. The initial solution x_i^0 was set to 0, the number of iterations was set to $N = 1000$, and no terminal condition was set. Figure 9 shows the convergence behavior, where the horizontal axis shows iterations, and the vertical axis shows the qualities of solutions as percentage of those obtained by LK. It is observed that the solution goes up and down, and gradually converges to a local minimum solution whose quality is 99.5% of the LK value.

SNR was also applied to 50 benchmark instances from TSPLIB [12], whose metric is Euclidean and whose optimal solutions are known. The sizes n of the instances range over 51 to 1000. The search was terminated in Step 14 when the best solution x^{best} was not updated during 100 consecutive iterations. The results obtained by SNR and LK are shown in Table I. The table shows that the results obtained by SNR have qualities

TABLE I

APPLICATION TO TSPLIB INSTANCES

(TOUR LENGTHS ARE EXPRESSED AS PERCENTAGES IN EXCESS OF THE OPTIMAL VALUES)

TSPLIB	SNR	LK	bier127	100.67	102.87	pr264	102.52	104.45
eil51	100.23	101.41	ch130	100.33	101.65	a280	101.40	101.51
berlin52	100.00	100.00	pr136	100.41	100.15	pr299	100.75	101.60
st70	100.59	101.04	pr144	101.29	100.09	lin318	101.97	101.35
eil76	101.67	100.19	ch150	100.55	100.25	linhp318	103.07	103.03
pr76	100.10	100.86	kroA150	100.16	100.96	rd400	102.74	102.66
rat99	100.50	101.16	kroB150	100.08	100.92	fl417	100.76	101.70
kroA100	100.05	100.00	pr152	100.18	100.84	pr439	105.45	100.82
kroB100	101.01	101.91	u159	101.10	100.00	pcb442	102.39	102.57
kroC100	100.50	104.08	rat195	102.54	101.38	d493	104.93	102.31
kroD100	100.32	101.61	d198	100.30	100.74	u574	101.12	102.11
kroE100	100.42	100.17	kroA200	100.96	103.97	rat575	102.63	102.51
rd100	100.43	100.00	kroB200	100.18	102.63	p654	103.48	102.59
eil101	100.79	101.27	pr226	100.30	100.05	d657	101.78	102.41
lin105	100.00	104.92	ts225	103.74	100.25	u724	104.59	103.22
pr107	100.00	100.30	tsp225	102.66	100.08	rat783	107.11	102.42
pr124	100.00	100.23	gil262	100.76	102.61	dsj1000	105.27	103.62
						Ave.	101.50	101.59

comparable to those found by LK, and that SNR converges to reasonably good solutions even when $n > M$.

5. Conclusion

A new function minimization algorithm called SNR was proposed and evaluated with quadratic functions and an objective function for a new TSP formulation based on parametric local search. The proposed algorithm is classified as a trust region method, where derivatives are approximated with sample points around a current solution, and a trust region is defined as a Gaussian distribution of unit variance. A unique property of the algorithm is that it allows the number of sample points to be smaller than the number of variables.

Through intensive numerical experiments, it has been shown that the described algorithm, SNR, has the following characteristics:

- It can approximate derivatives with fewer sample points than the dimension of the objective function.

- It can converge to reasonably good solutions with inexact derivative information even when the objective landscape is noisy.

References

[1] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *T*he Computer Journal, Vol. 7, pp. 155-161, 1964.

[2] J. A. Nelder and R. Mead, "A simplex method for function minimizatoin," *T*he Computer Journal, Vol. 7, pp. 308-313, 1965.

[3] J. E. Dennis and V. Torczon, "Direct search methods on parallel machines," *S*IAM Journal on Optimization, Vol. 1, pp. 448-474, 1991.

[4] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," *U*niversity of Cambridge Numerical Analysis Reports, DAMTP 1992/NA5, 1992.

[5] A. R. Conn, K. Scheinberg, and Ph. L. Toint, "Recent progress in unconstrained nonlinear optimization without derivatives," *M*athematical Programming, Vol. 79, pp. 397-414, 1997.

[6] E. A. Novikov, "Functionals and the random-force method in turbulence theory," *S*oviet Physics JETP, Vol. 20, pp. 1290-1294, 1965.

[7] M. Koda and H. Okano, "A New Stochastic Learning Algorithm for Neural Networks," *J*ournal of Operations Research Society of Japan, Vol. 43, pp. 469-485, 2000.

[8] H. Okano and M. Koda, "A New Noise-Based Gradient Method and Its Applications," *I*BM Research Report, RT0390, 2000.

[9] D. K. Dacol and H. Rabitz, "Sensitivity Analysis of Stochastic Kinetic Models," *J*ournal of Mathematical Physics, Vol. 25, pp. 2716-2727, 1984.

[10] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative Studies," *J*ournal of Statistic Physics, Vol. 34, pp. 975-986, 1984.

[11] S. Lin and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *O*perations Research, Vol. 21, pp. 498-516, 1973.

[12] G. Reinelt, "TSPLIB–A Traveling Salesman Problem Library," *O*RSA Journal on Computing, Vol. 3, pp. 376-384, 1991.